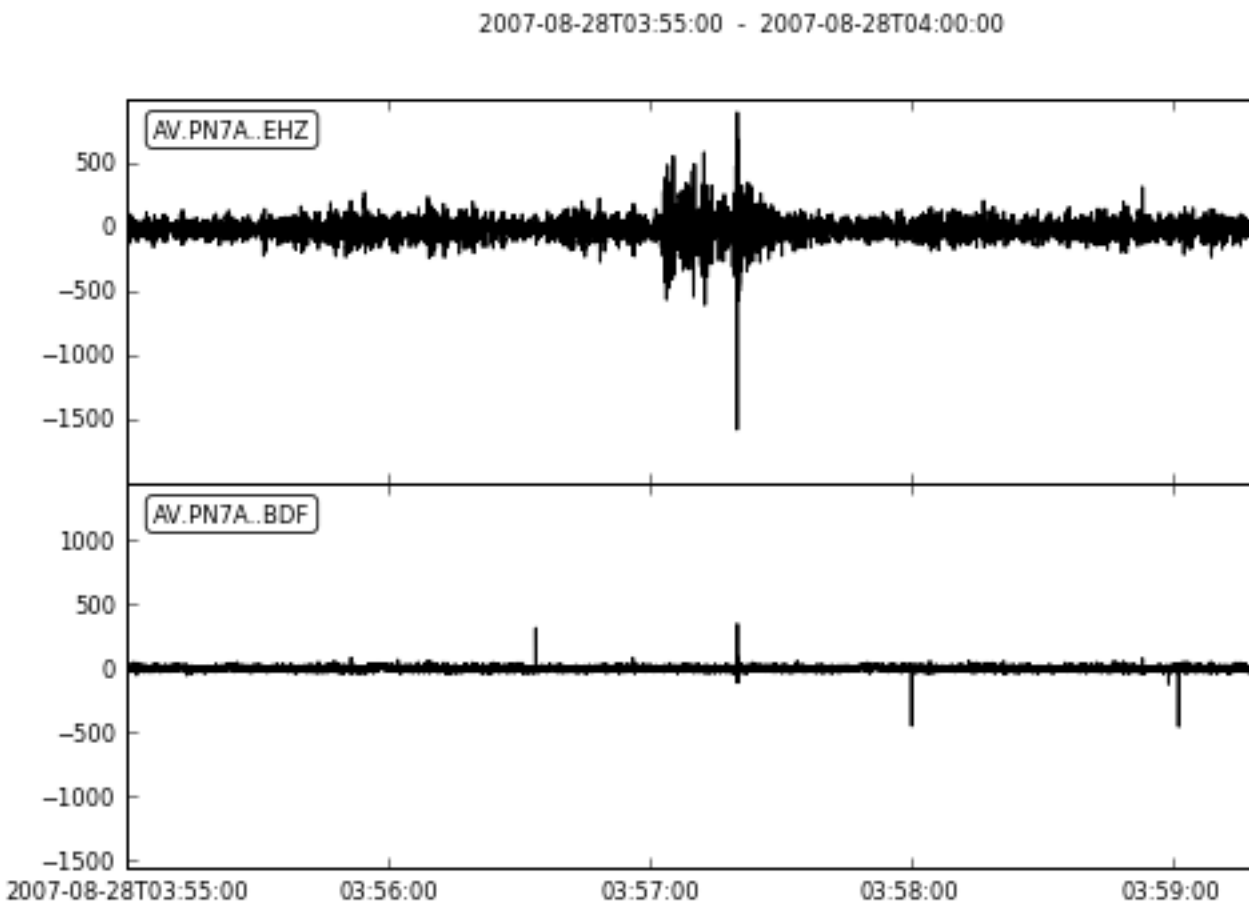


In [11]:

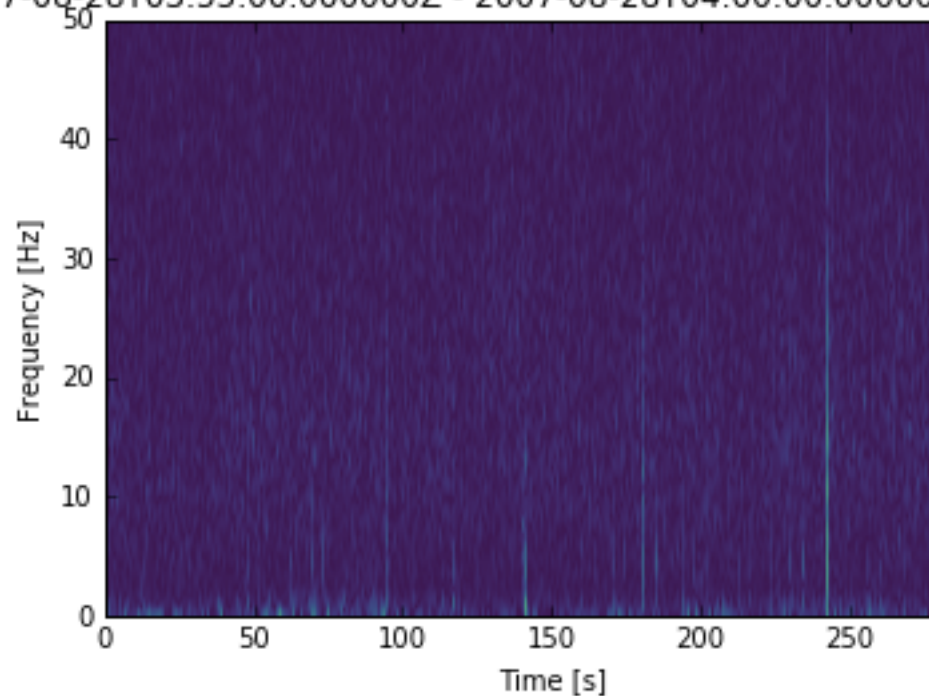
In [11]:

In [12]: `import sys`

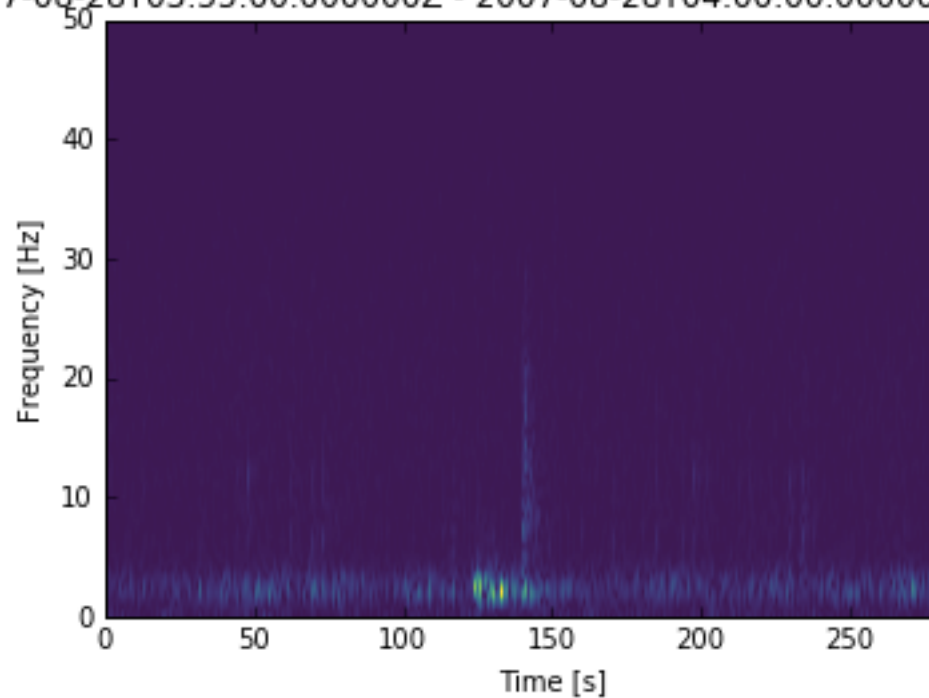
```
....:
sys.path.append('/Users/glennthompson/Dropbox/scratch_matlab')
....: import tune_sta_lta as tsl
....: from obspy.core import read
....: from obspy.core.utcdatetime import UTCDateTime
....: import obspy.signal.trigger as trigger
....: tstart = UTCDateTime(2007, 8, 28, 3, 55, 0)
....: tend = UTCDateTime(2007, 8, 28, 4, 0, 0)
....: st =
read("/Users/glennthompson/Dropbox/scratch_matlab/SEEDDATA/PN7A.*
.2007.240", starttime=tstart, endtime=tend)
....: st.plot()
....: st.spectrogram()
```



AV.PN7A..BDF | 2007-08-28T03:55:00.000000Z - 2007-08-28T04:00:00.000000

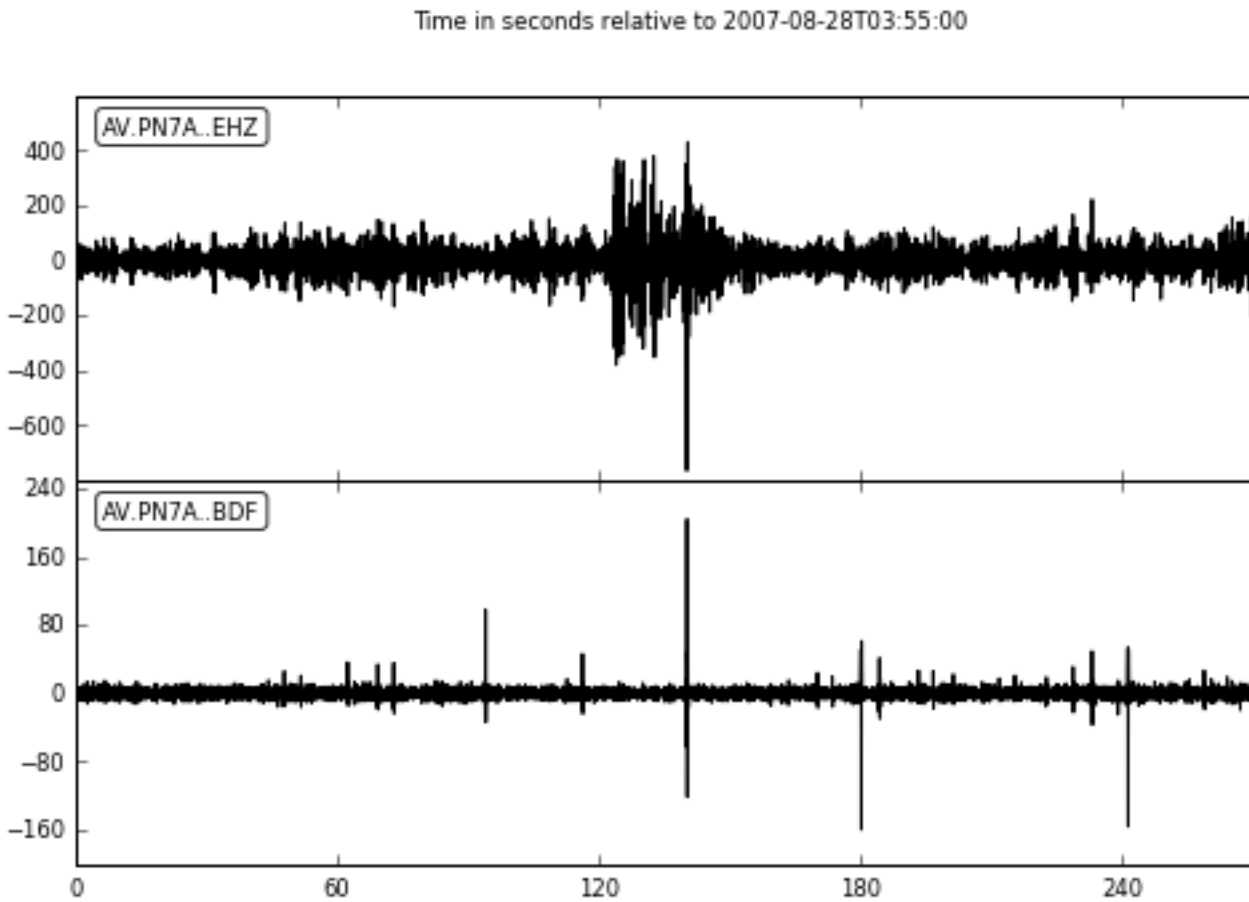


AV.PN7A..EHZ | 2007-08-28T03:55:00.000000Z - 2007-08-28T04:00:00.000000

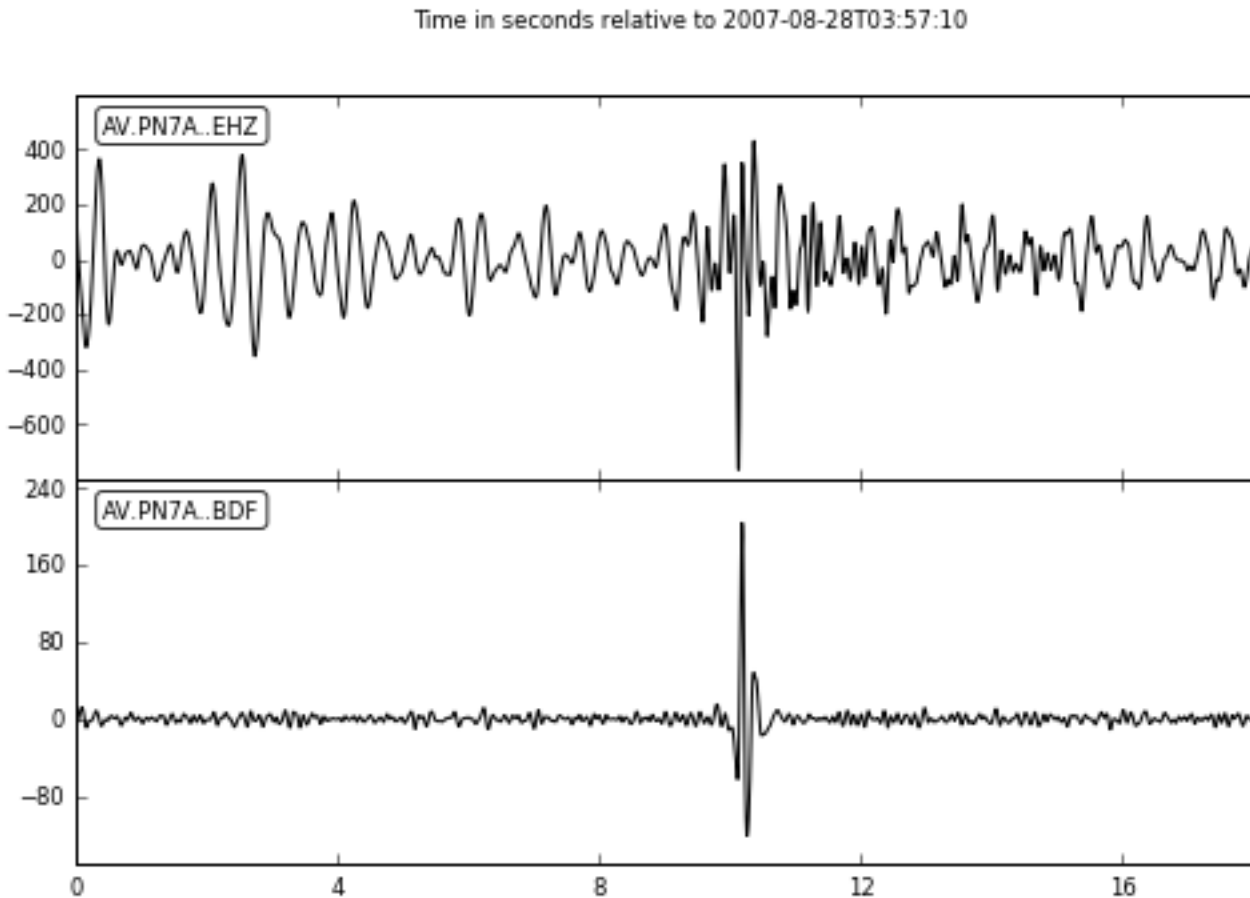


Out[12]: [None, None]

```
In [13]: st.filter('bandpass', freqmin=2.0, freqmax=10.0,
...: st.plot(type='relative', equal_scale=False)
```



```
In [14]: t_signal_start = 130.0
...: t_signal_end = 150.0
...: st.plot(type='relative', equal_scale=False,
starttime=tstart+t_signal_start, endtime=tstart+t_signal_end)
...:
```



```
In [15]: algorithm = 'classic_sta_lta'
...: numtries = 100
...: tr_infrasound = st[0]
...: tr_seismic = st[1]
...: result = tsl.tune_sta_lta(tr_infrasound, algorithm,
t_signal_start, t_signal_end, numtries )
...: sta_best = result[0]
...: lta_best = result[1]
...: print "Best STA window = %.1f seconds, Best LTA window =
%.1f seconds" % (sta_best, lta_best)
...:
```

```
Algorithm: classic_sta_lta
sta_seconds=2.4 lta_seconds=12.0 max(staltaratio)=4.8
sta_seconds=5.7 lta_seconds=34.2 max(staltaratio)=5.0
sta_seconds=6.1 lta_seconds=48.8 max(staltaratio)=6.3
sta_seconds=4.7 lta_seconds=37.6 max(staltaratio)=6.6
sta_seconds=1.3 lta_seconds=10.4 max(staltaratio)=7.7
sta_seconds=3.1 lta_seconds=27.9 max(staltaratio)=7.7
sta_seconds=2.0 lta_seconds=18.0 max(staltaratio)=8.4
```

```

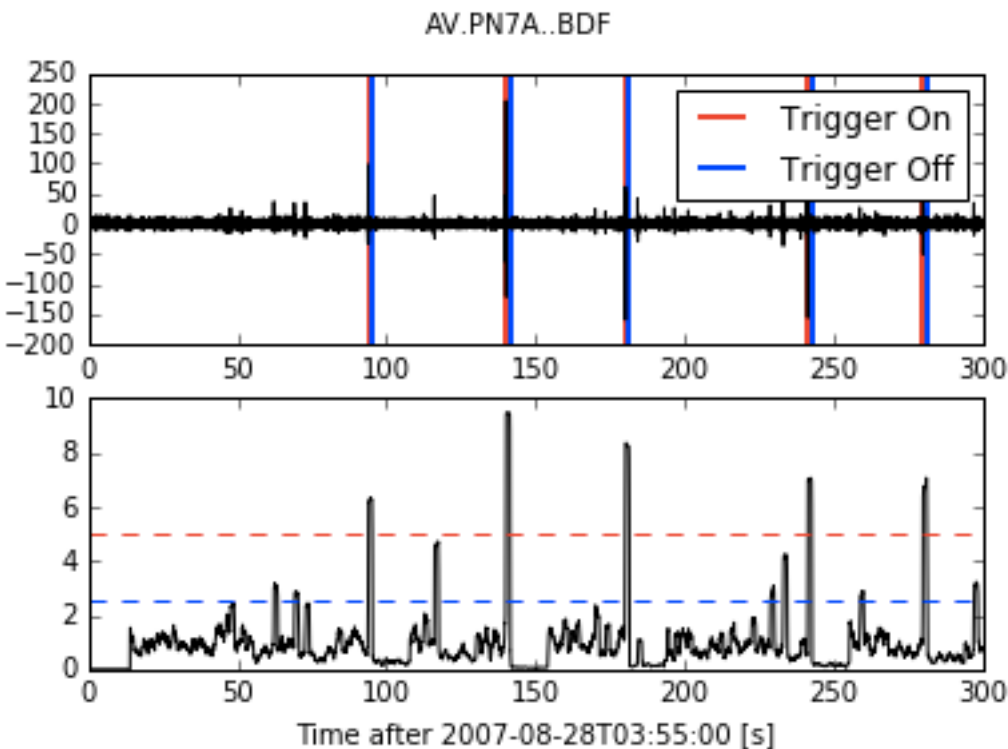
sta_seconds=1.0 lta_seconds=9.0 max(staltaratio)=8.7
sta_seconds=0.3 lta_seconds=2.7 max(staltaratio)=8.9
sta_seconds=1.4 lta_seconds=14.0 max(staltaratio)=9.5
Best STA window = 1.4 seconds, Best LTA window = 14.0 seconds

```

```

In [16]: staltaratio_best = result[2]
        ....: thresh_on = 5
        ....: thresh_off = 2.5
        ....: trigger.plot_trigger(tr_infrasound, staltaratio_best,
        thresh_on, thresh_off)

```



```

In [17]: triggers_per_event = 1 # set this to 2 and it would have
to trigger on both channels to declare event, 1 means trigger on
either is okay
        ....: import re # for some dumb reason, coincidence trigger
needs algorithm name without the underlines!
        ....: algorithm_without_underlines = re.sub('_', '',
algorithm)
        ....: trig =
trigger.coincidence_trigger(algorithm_without_underlines,
thresh_on, thresh_off, st, triggers_per_event, sta=sta_best,
lta=lta_best)
        ....: from pprint import pprint
        ....: pprint(trig)

```

```

    ....: print "Number of events detected = %d" % len(trig)
[{'u'coincidence_sum': 1.0,
  u'duration': 1.3999998569488525,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 56, 33, 930000),
  u'trace_ids': [u'AV.PN7A..BDF']},
 {'u'coincidence_sum': 1.0,
  u'duration': 2.8299999237060547,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 57, 3, 700000),
  u'trace_ids': [u'AV.PN7A..EHZ']},
 {'u'coincidence_sum': 1.0,
  u'duration': 1.5300002098083496,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 57, 20, 120000),
  u'trace_ids': [u'AV.PN7A..BDF']},
 {'u'coincidence_sum': 1.0,
  u'duration': 1.4200000762939453,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 58, 0, 210000),
  u'trace_ids': [u'AV.PN7A..BDF']},
 {'u'coincidence_sum': 1.0,
  u'duration': 1.4000000953674316,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 59, 1, 500000),
  u'trace_ids': [u'AV.PN7A..BDF']},
 {'u'coincidence_sum': 1.0,
  u'duration': 1.4600000381469727,
  u'similarity': {},
  u'stations': [u'PN7A'],
  u'time': UTCDateTime(2007, 8, 28, 3, 59, 40, 120000),
  u'trace_ids': [u'AV.PN7A..BDF']}]
Number of events detected = 6

```

```

In [18]: pretrig = 5;
    ....: posttrig = 5;
    ....: count = 0
    ....: for thistrig in trig:
    ....:     count += 1
    ....:     print "Event %d at %s" %
(count, thistrig['time'].isoformat())

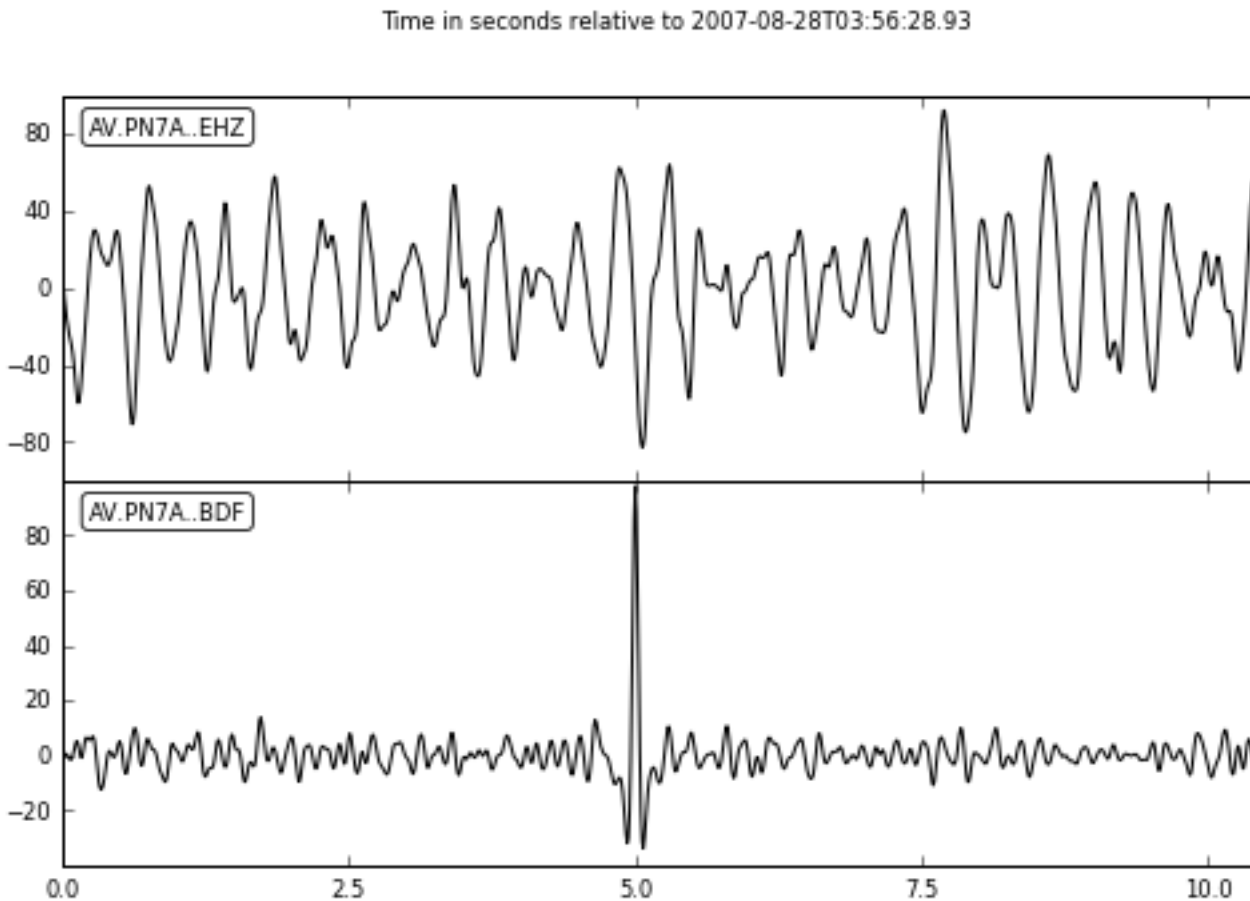
```

```

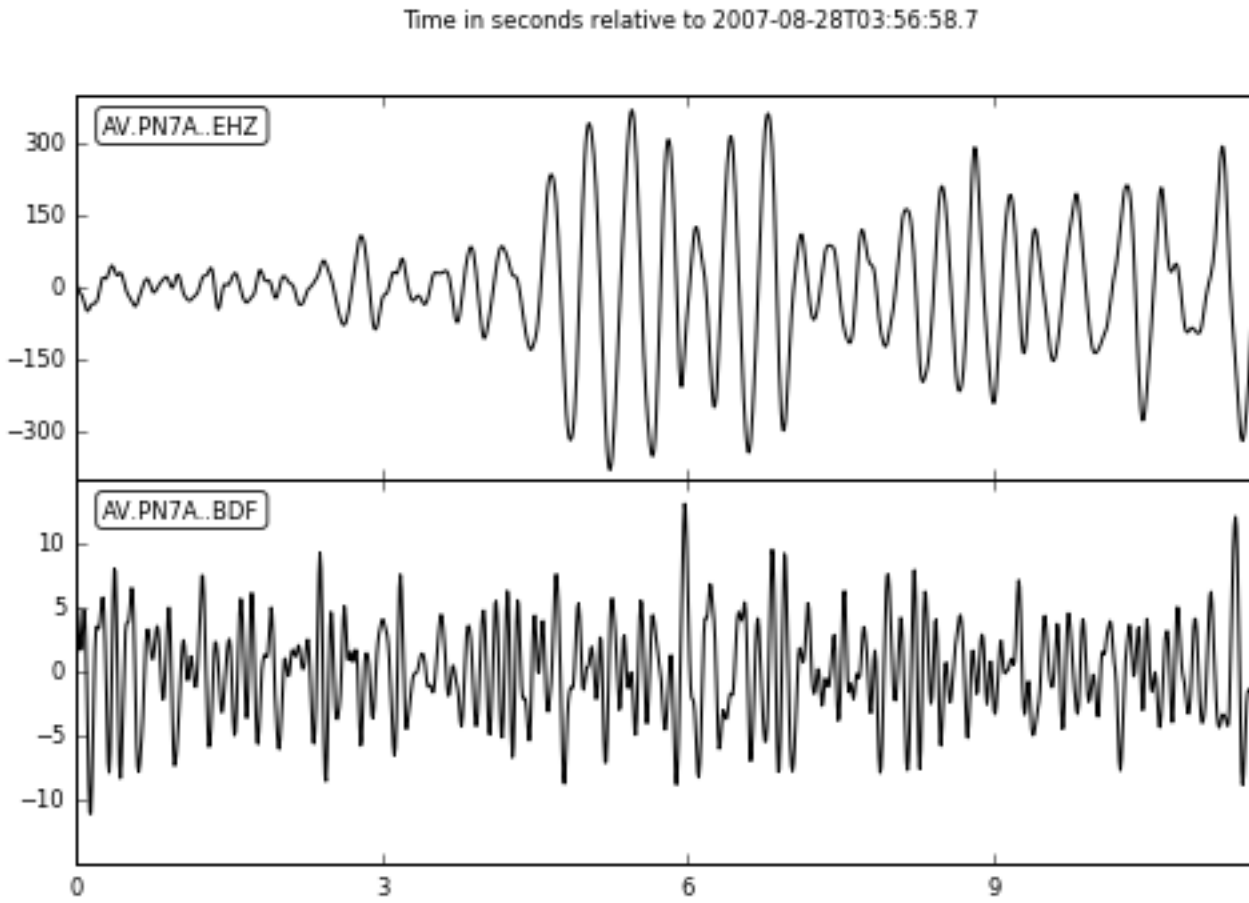
....: st2 = st.copy()
....: st2.trim(starttime = thistrig['time'] - pretrig,
endtime = thistrig['time'] + thistrig['duration'] + posttrig)
....: st2.plot(type='relative', equal_scale=False)
....:
....:

```

Event 1 at 2007-08-28T03:56:33.930000

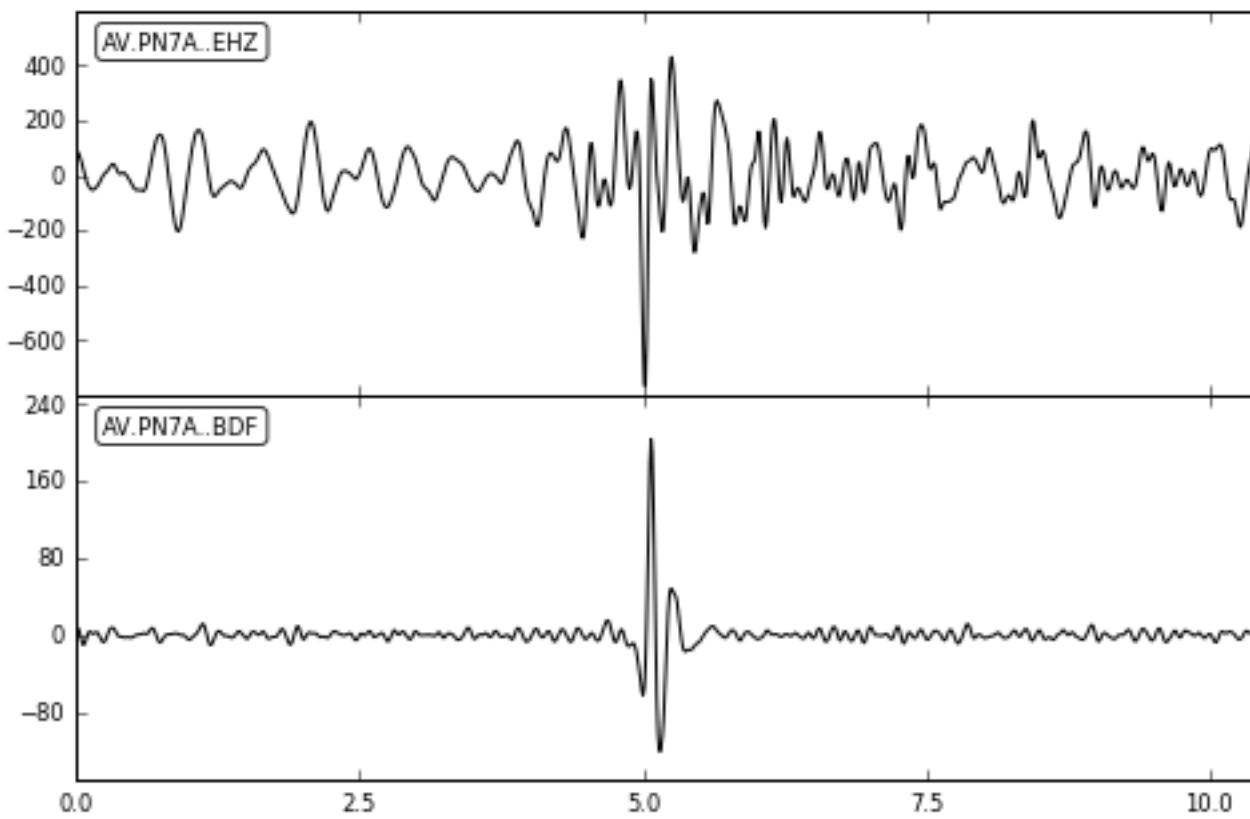


Event 2 at 2007-08-28T03:57:03.700000



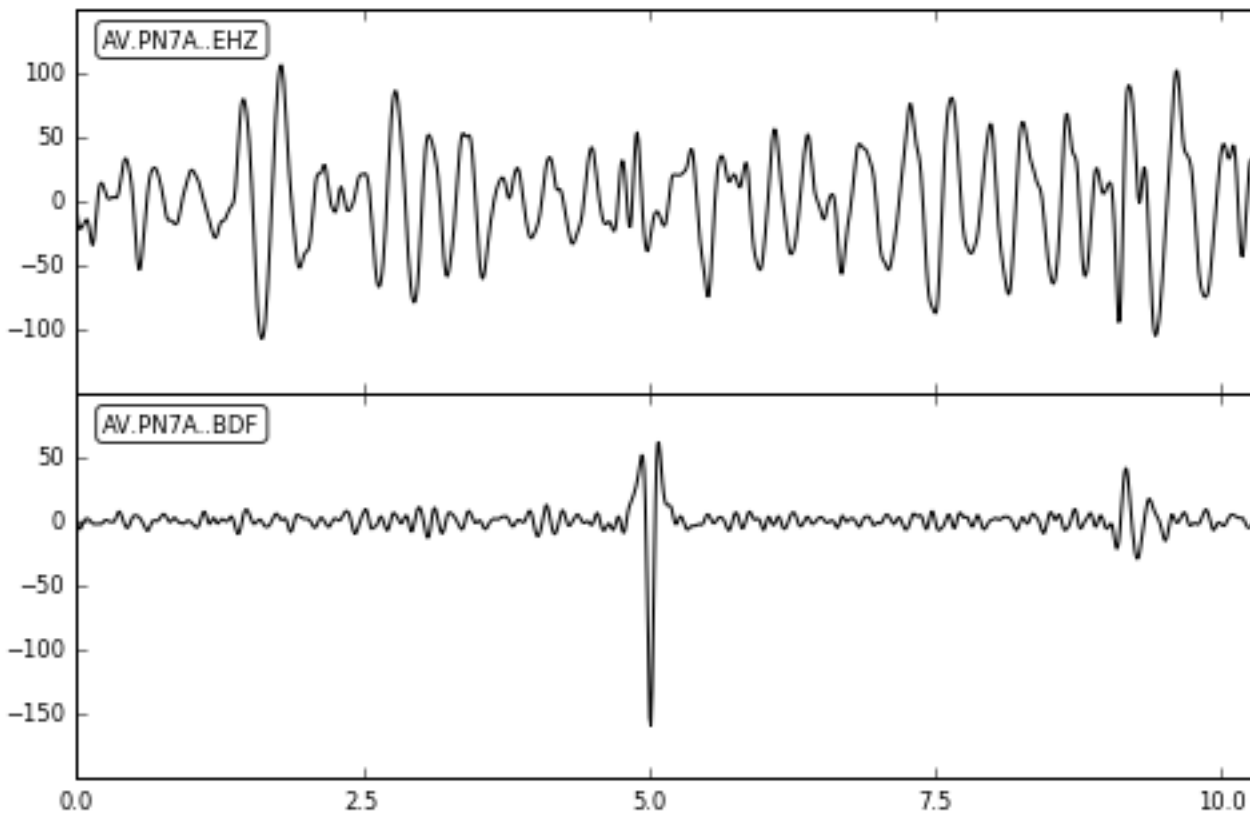
Event 3 at 2007-08-28T03:57:20.120000

Time in seconds relative to 2007-08-28T03:57:15.12

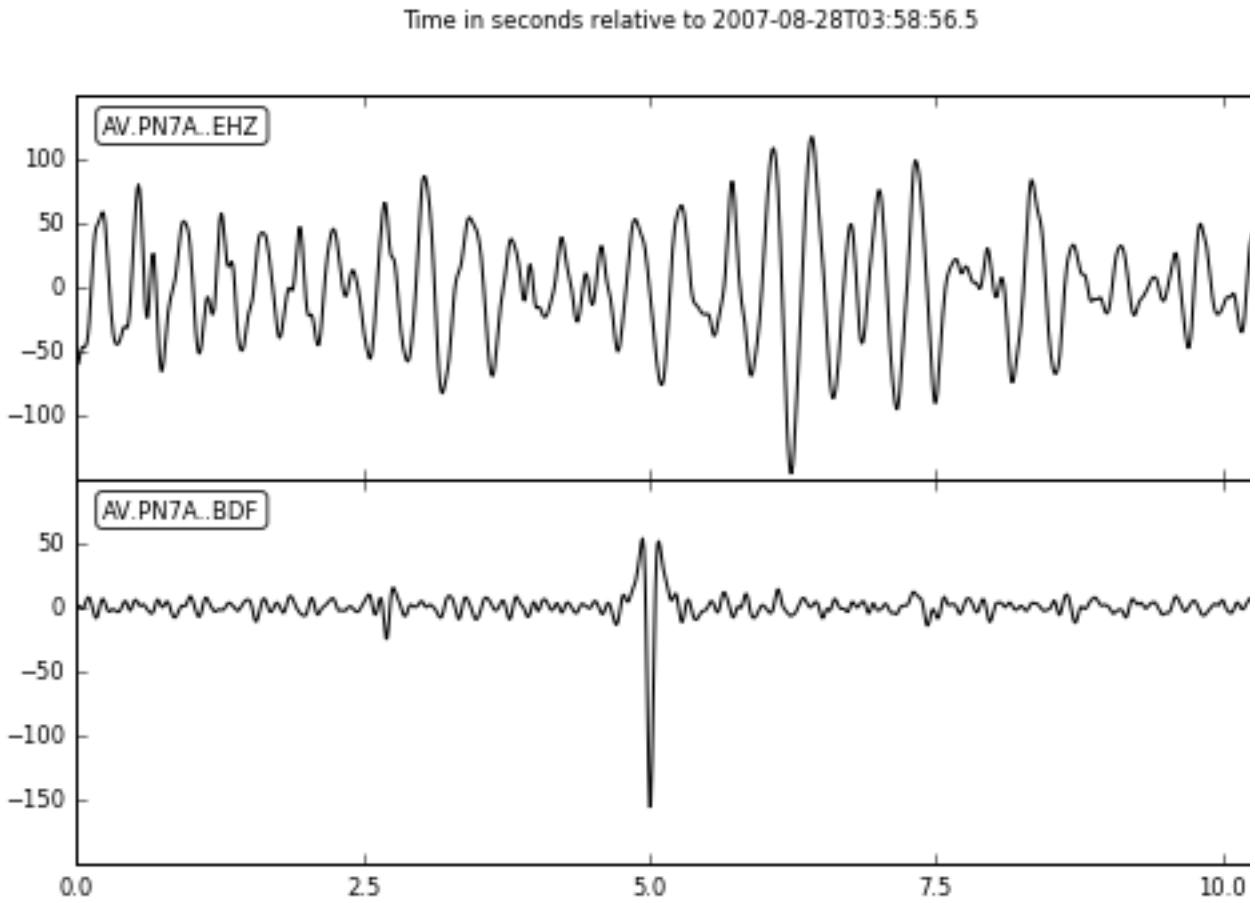


Event 4 at 2007-08-28T03:58:00.210000

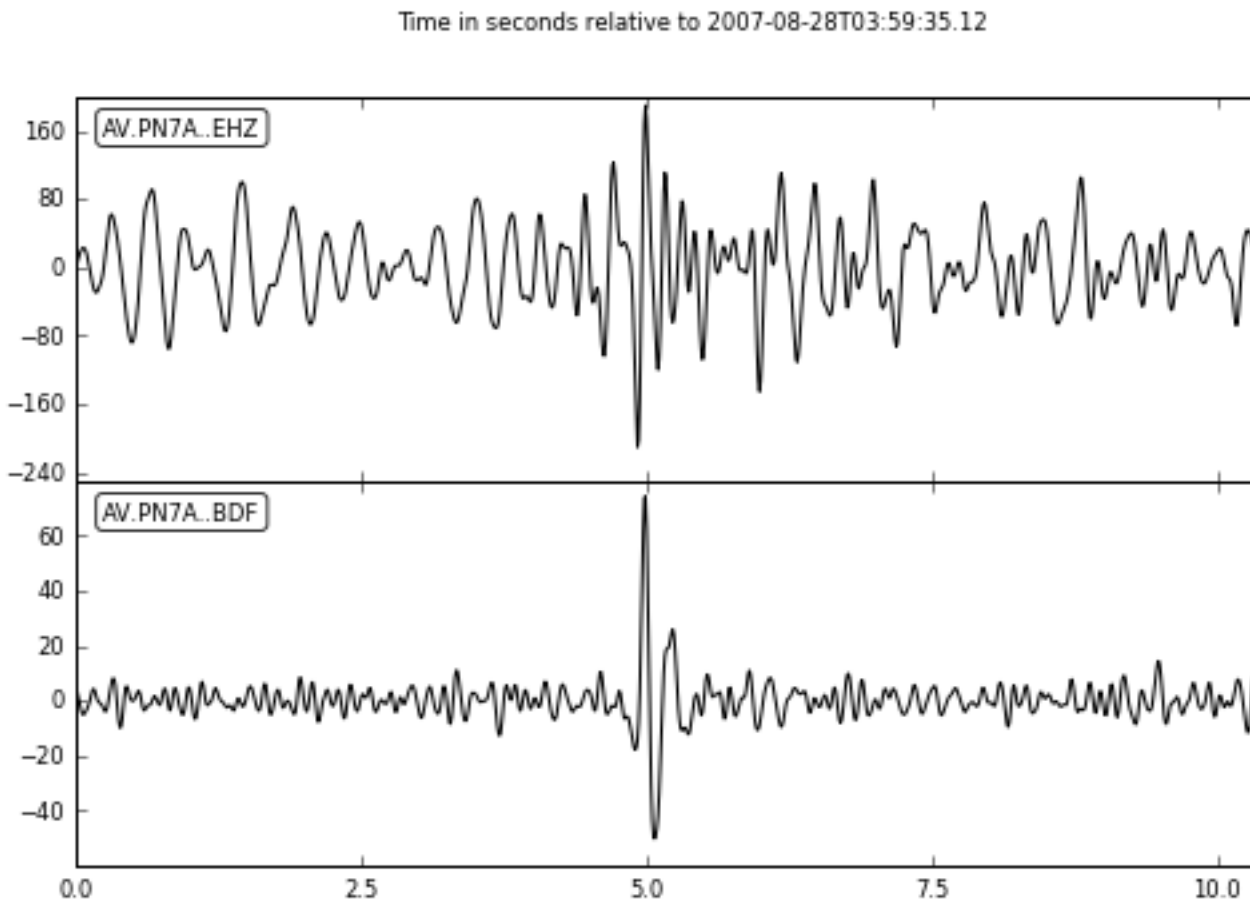
Time in seconds relative to 2007-08-28T03:57:55.21



Event 5 at 2007-08-28T03:59:01.500000



Event 6 at 2007-08-28T03:59:40.120000



In [19]: